Steps for hypothesis testing

Step 0: Define problem Objective

Step 1: Specify the Null Hypothesis.

Step 2: Specify the Alternative Hypothesis.

Step 3: Set the Significance Level

Step 4: Calculate the Test Statistic and Corresponding P-Value.

Step 5: Drawing a Conclusion.

In [ ]:

Step 0: Define problem Objective:

Exercising does not affect weight`

This statement is my Hypothesis. Let's call it Null hypothesisfor now. For now, it is the status quo as in we consider it to be true

Step 1 and 2: Specify Null and Alternate Hypothesis:

H^0: Exercising does not affect weight. Or equivalently $\mu = 0$

H^A: Exercise does reduce weight. Or equivalently $\mu > 0$

We collect weight loss data for a sample of 30 people who regularly exercise for over 3 months.

- WeightLoss Sample Mean = 2 kg
- Sample Standard Deviation = 1 kg

Does this prove that exercise does reduce weight? Infact, it sort of looks like that exercising does have its benefits as people who exercise have lost on an average 2 kgs.

Step 3: Set the Significance Level

Assuming that the null hypothesis is true, what is the probability of observing a sample mean of 2 kg or more extreme than 2 kg?

- Assuming we can calculate this — If this probability value is meagre (lesser than a threshold value), we reject our null hypothesis. And otherwise, we fail to reject our null hypothesis. Why fail to reject and not accept?
- This probability value is actually the p-value. Simply, it is just the probability of observing what

In [ ]:

Step 4 Calculate the test statastics and corresponding p-value

In [1]:
```
from scipy.stats import norm
import numpy as np
```

In [2]:
```
### loc is the location parameter and mean is the normal distribution
p=1-norm.cdf(x=2,loc=0,scale=1/np.sqrt(30))
p
```

Out[2]: 0.0

As the p value(0.0)is less than significant level(0.05) therefore we have to reject the null hypothesis(Exercising does not affect weight.)

Step 5: Drawing the Conclusion

As such, this is a very small probability p-value (less than the significance level of 0.05) for the mean of a sample to take a value of 2 or more.

And so we can reject our Null hypothesis. And we can call our results statistically significant as in they don't just occur due to mere chance

In [3]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
cars_df=pd.read_csv("cars93.csv")
```

In [5]:
```
cars_df.head()
```

Out[5]:

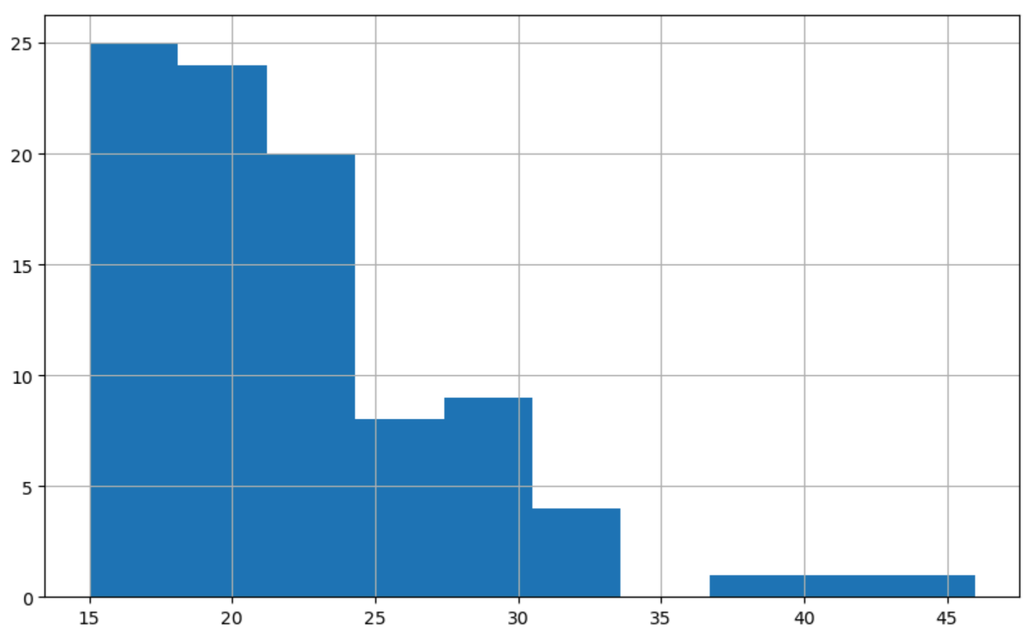
	Manufacturer	Model	Type	Price	MPG.city	AirBags	Horsepower	Passengers	Rear.seat.room	Luggage.room
0	Acura	Integra	Small	15.9	25	Driver only	140	5	26.5	11.0
1	Acura	Legend	Midsize	33.9	18	Driver & Passenger	200	5	30.0	15.0
2	Audi	90	Compact	29.1	20	Driver only	172	5	28.0	14.0
3	Audi	100	Midsize	37.7	19	Driver & Passenger	172	6	31.0	17.0
4	BMW	535i	Midsize	30.0	22	Driver only	208	4	27.0	13.0

In [9]:
```
cars_df.shape
```

Out[9]: (93, 10)

In [6]:
```
cars_df["MPG.city"].hist(figsize=(10,6),bins=10)
```

Out[6]: <Axes: >



In above graph we can see that the data is right skweed

So for proving central limit theorem we will collect random sampels from the data and plot their mean

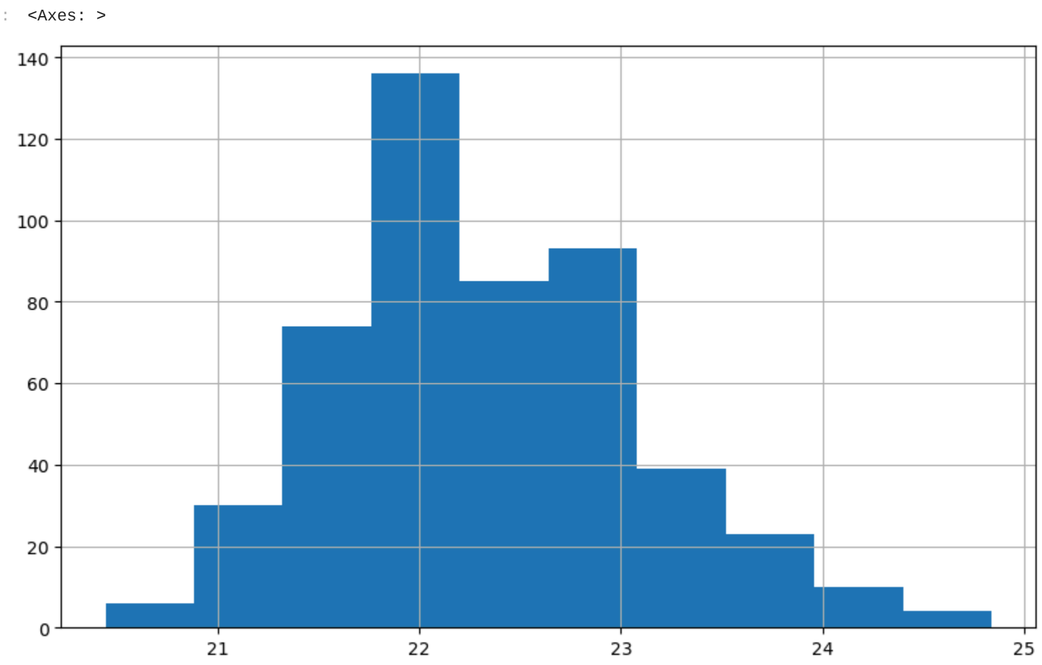
In [7]:
```
sampling_distribution=[]

for i in range(500):
    #### generates random 50 integer index between 0 to len ("MPG.city")
    sample_index=np.random.randint(0,len(cars_df["MPG.city"]),50)
    ### Extract
    sample=cars_df["MPG.city"][sample_index]
    sample_mean=sample.mean()
    sampling_distribution.append(sample_mean)
print(sampling_distribution)
```

```
[21.44, 21.94, 23.58, 21.9, 22.38, 21.88, 22.7, 22.4, 21.96, 22.14, 22.14, 23.6, 23.68, 22.06, 21.46, 22.0, 21.6, 23.78, 21.02, 22.64, 22.94, 22.76, 23.16, 21.6, 22.02, 22.78, 23.08, 22.48, 23.0, 21.64, 21.18, 24.06, 21.58, 21.7, 22.4,
22.52, 21.96, 22.48, 21.36, 21.48, 22.98, 22.52, 21.72, 21.56, 21.48, 21.6, 22.18, 23.8, 21.86, 22.22, 20.98, 21.86, 22.74, 23.02, 21.9, 22.22, 21.72, 21.82, 21.76, 22.28, 23.02, 21.94, 22.14, 21.46, 23.16, 22.2, 21.76, 23.92, 22.18, 2
0.6, 22.64, 23.22, 22.02, 21.96, 21.98, 23.66, 21.9, 22.94, 22.6, 21.6, 22.8, 22.24, 22.12, 21.9, 22.12, 21.84, 22.16, 22.1, 22.5, 22.02, 21.14, 21.56, 21.08, 23.4, 22.66, 21.44, 21.74, 22.8, 21.7, 21.66, 22.3, 21.4, 21.78, 20.66, 20.4
4, 22.56, 21.84, 22.18, 21.68, 22.6, 21.16, 22.28, 21.96, 21.58, 23.34, 22.1, 22.94, 21.18, 21.54, 22.2, 22.88, 23.44, 22.88, 23.2, 23.74, 21.9, 22.46, 22.08, 22.08, 22.08, 21.5, 22.08, 24.62, 23.74, 22.26, 22.98, 22.06, 24.32, 21.58,
22.54, 21.22, 21.96, 23.94, 23.34, 22.94, 22.58, 22.12, 21.9, 22.14, 23.02, 22.08, 21.12, 21.02, 22.7, 21.26, 23.5, 21.76, 22.06, 23.14, 22.24, 22.48, 21.3, 21.22, 21.78, 23.1, 22.78, 21.48, 22.74, 22.1, 21.74, 22.74, 22.2, 22.68, 22.1
8, 23.58, 21.74, 21.4, 21.66, 21.82, 21.92, 22.74, 22.54, 22.2, 22.12, 23.24, 22.68, 22.86, 22.84, 21.68, 22.94, 21.98, 22.1, 22.84, 21.48, 22.08, 22.64, 21.08, 21.78, 22.62, 21.68, 21.1, 22.98, 21.98, 21.86, 22.72, 22.22, 22.2, 21.68,
21.62, 21.94, 22.28, 22.68, 22.3, 22.7, 23.1, 22.08, 20.92, 24.44, 22.58, 21.7, 22.16, 24.84, 21.58, 21.64, 22.64, 21.18, 22.78, 22.34, 22.16, 23.64, 23.42, 22.62, 23.58, 22.06, 21.48, 22.62, 22.66, 22.7, 23.04, 21.82, 22.82, 22.9, 22.
12, 21.78, 22.44, 21.12, 21.54, 21.82, 22.52, 22.86, 21.9, 22.0, 22.72, 23.14, 22.4, 23.24, 21.88, 22.34, 21.88, 22.5, 21.58, 21.68, 20.76, 22.04, 23.2, 22.84, 22.28, 21.92, 23.4, 22.42, 21.7, 23.6, 22.76, 23.6, 22.02, 22.62, 21.86, 2
4.12, 23.22, 23.18, 21.78, 22.68, 22.84, 23.68, 22.04, 22.96, 23.56, 22.3, 22.12, 22.8, 24.38, 23.38, 22.46, 22.12, 22.54, 22.98, 24.16, 22.36, 21.44, 22.56, 21.8, 21.18, 23.34, 21.98, 23.2, 22.92, 20.96, 22.74, 23.64, 22.12, 21.46, 2
1.6, 22.66, 22.96, 22.92, 22.0, 22.52, 22.52, 22.12, 21.72, 23.0, 23.1, 22.7, 21.88, 22.58, 21.66, 21.9, 22.38, 22.92, 22.2, 22.88, 21.88, 21.5, 21.92, 21.76, 22.1, 22.94, 21.3, 23.04, 21.86, 23.38, 22.3, 21.98, 22.3, 22.58, 21.82, 21.
94, 21.58, 22.24, 22.48, 23.16, 21.56, 21.72, 21.9, 23.34, 22.04, 21.88, 21.74, 21.6, 22.4, 22.14, 22.06, 23.06, 22.02, 21.14, 22.04, 22.58, 22.08, 23.22, 20.76, 21.62, 21.44, 21.0, 21.88, 21.34, 22.24, 22.24, 21.74, 21.48, 21.18, 21.0
8, 22.68, 23.28, 22.0, 22.12, 23.16, 21.78, 21.04, 21.84, 23.02, 22.06, 21.32, 21.34, 22.82, 21.5, 22.94, 24.4, 22.72, 23.96, 22.1, 22.36, 21.56, 21.24, 21.38, 22.9, 22.88, 21.96, 22.0, 24.2, 22.34, 22.74, 22.68, 22.86, 21.96, 24.38, 2
1.26, 22.22, 22.02, 22.68, 22.8, 21.44, 21.64, 23.14, 21.82, 23.88, 21.62, 20.5, 23.6, 22.32, 22.74, 22.2, 22.24, 22.8, 21.68, 21.98, 22.76, 22.82, 22.78, 22.18, 21.9, 22.64, 22.74, 22.24, 23.16, 23.52, 22.1, 22.14, 23.32, 22.16, 23.1,
24.1, 23.06, 22.5, 21.28, 23.1, 22.02, 21.96, 22.14, 21.96, 21.84, 21.54, 22.82, 22.04, 21.4, 24.34, 22.4, 22.84, 22.44, 22.38, 22.52, 22.28, 23.62, 22.56, 23.28, 23.48, 21.2, 22.12, 21.5, 23.52, 22.76, 23.0, 22.42, 22.6, 22.08, 22.52,
21.5, 22.8, 22.9, 22.42, 22.22, 21.86, 22.1, 22.02, 22.5, 23.1, 21.94, 22.02, 22.38, 22.78, 22.9]
```

In this above cell we declared one empty dictionary ie sampling_distribution which holds random samples,in for loop we are generating random numbers within range of 0 to length of our data and after we are taking the mean of those sample. After taking mean we are appending that value in our empty list

In [8]:
```
pd.Series(sampling_distribution).hist(figsize=(10,6),bins=10)
```

Out[8]: <Axes: >



Now as we can see the distribution of sample's mean it is in proper bell shape which means the data is normally distributed ie Mean=Mode=Median

In [ ]: